

**Кайранбаева Альбина Бериккалиевна\*<sup>1</sup>,**

магистр технических наук, старший преподаватель ЧВПОУ Западно-Казахстанский  
инновационно-технологический университет, Республика Казахстан, 090006,  
г. Уральск, [a.center.t@mail.ru](mailto:a.center.t@mail.ru), ORCID ID: 0009-0001-4477-1136

## **АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА**

**Аннотация.** В данной статье рассматривается использование методов искусственного интеллекта (ИИ) в автоматизации тестирования программного обеспечения. В условиях постоянного усложнения программных систем и сокращения сроков поставки продуктов, традиционные подходы к тестированию становятся менее эффективными. Применение ИИ открывает новые возможности для оптимизации процессов тестирования, повышения их гибкости и снижения затрат. В работе проанализированы основные технологии, включая машинное обучение, нейросетевые архитектуры, обработку естественного языка и интеллектуальные системы поддержки принятия решений. Особое внимание уделено современным инструментам автоматизированного тестирования с элементами ИИ, таким как Testim.io, Applitools, Functionize и Mabl. Описаны преимущества интеграции ИИ в QA: автоматическая генерация тестов, адаптация к изменениям, выявление сложных дефектов и расширение покрытия. Вместе с тем освещены актуальные вызовы: проблема «чёрного ящика», требования к качеству данных, стоимость внедрения и поддержка нестабильных интерфейсов. В заключении представлены перспективы развития, включая использование генеративных моделей, Explainable AI и автономных тестирующих агентов. Проведённый обзор демонстрирует, что внедрение ИИ становится неотъемлемой частью современной практики обеспечения качества, повышая эффективность и интеллектуальный уровень тестирования.

**Ключевые слова:** искусственный интеллект, автоматизация тестирования, машинное обучение, нейросети, интеллектуальные системы, software testing, AI in QA.

**Введение.** С развитием программной инженерии и ростом требований к качеству цифровых продуктов возросла необходимость в надёжных и эффективных методах тестирования. Ручные методы становятся всё менее эффективными из-за их затратности, зависимости от человеческого фактора и ограниченной масштабируемости. Стандартная автоматизация тестов частично решает эти проблемы, но сталкивается с трудностями в условиях быстро изменяющихся требований, частых релизов и высокой вариативности пользовательского поведения [1,2].

На этом фоне всё чаще обсуждается потенциал искусственного интеллекта — совокупности технологий, способных анализировать данные, обучаться на них и принимать решения. В контексте тестирования ИИ способен не просто выполнять тестовые сценарии, но и самостоятельно их генерировать, адаптировать к изменениям, выявлять аномалии и даже прогнозировать области, в которых, вероятно, возникнут дефекты [2].

Кроме того, ИИ позволяет устранить так называемые «узкие места» в тестировании — такие как неполное покрытие, избыточность тестов и невозможность анализа всей документации вручную. Использование технологий NLP, например, позволяет

автоматически извлекать требования из текстов и генерировать на их основе тест-кейсы, а машинное обучение — выявлять шаблоны в баг-репортах, предсказывать уязвимые компоненты и оптимизировать регрессионное тестирование [2,8].

Таким образом, задача статьи — исследовать современные методы применения ИИ в автоматизированном тестировании, оценить их эффективность, определить актуальные инструменты и возможные риски. Кроме того, важно понять, каковы перспективы и насколько ИИ готов к повсеместному внедрению в процессы обеспечения качества [2].

Теоретические основы автоматизированного тестирования с использованием ИИ [2].

Автоматизированное тестирование с ИИ основывается на применении интеллектуальных алгоритмов к задачам контроля качества программного обеспечения [3]. Это направление сочетает в себе классические методы QA с современными достижениями в области машинного обучения, анализа больших данных и нейросетевых вычислений. Основной целью является повышение точности, скорости и адаптивности тестирования [2].

**Машинное обучение (ML)** в тестировании применяется для построения моделей, способных выявлять дефекты на основе исторических данных. Такие модели анализируют данные о прошлых ошибках, коммитах, изменениях кода и метриках, чтобы предсказывать наиболее уязвимые модули. Кроме того, ML помогает в кластеризации багов, генерации тест-кейсов, классификации логов и выявлении дубликатов отчётов об ошибках [2].

**Глубокое обучение (DL)**, разновидность ML, применяется для сложных задач, таких как визуальное тестирование. С помощью сверточных нейросетей можно находить различия между скриншотами разных версий интерфейса, даже если они неочевидны для обычных инструментов. DL также используется для распознавания элементов интерфейса, генерации пользовательских сценариев и автоматического заполнения форм [2, 4].

**Нейронные сети**, в частности рекуррентные (RNN) и трансформеры, используются для анализа логов, предсказания сбоев и автогенерации шагов тестов. Эти сети способны «обучаться» на последовательностях действий и предсказывать вероятное поведение пользователя или возможные сбои в системе [2].

**Обработка естественного языка (NLP)** применяется для анализа документации, требований и тикетов. Технологии NLP могут извлекать ключевые сущности из текста, определять бизнес-логику и даже сопоставлять требования с существующими тест-кейсами, предлагая дополнения или автоматическую генерацию новых тестов [2,5].

Существуют и гибридные подходы, сочетающие сразу несколько методов. Например, система может использовать NLP для понимания требований, ML — для анализа истории багов, а DL — для тестирования визуальной части интерфейса. Таким образом, ИИ в тестировании — это не один алгоритм, а целая экосистема технологий, каждая из которых решает конкретную задачу [2].

**Современные технологии и инструменты.** Развитие технологий ИИ способствовало появлению специализированных инструментов, способных автоматизировать различные аспекты тестирования. Эти решения сочетают в себе возможности анализа данных, машинного обучения, NLP и компьютерного зрения. Рассмотрим наиболее популярные и функционально развитые платформы, применяющие ИИ в QA [2].

**Testim.io** — одна из первых платформ, интегрировавших ИИ в процесс тестирования пользовательского интерфейса. Она использует машинное обучение для анализа поведения пользователей и создания стабильных автотестов. Благодаря этому

уменьшается количество ложных срабатываний и тесты адаптируются к изменениям в DOM-структуре страниц. Кроме того, Testim предлагает визуальный редактор, упрощающий работу специалистов без навыков программирования [2,5].

**Applitools Eyes** представляет собой инструмент визуального тестирования, основанный на компьютерном зрении и глубоких нейросетевых архитектурах. Он способен выявлять мельчайшие визуальные изменения между экранами различных версий приложения, даже если они не затрагивают функциональность. Это особенно важно для обеспечения консистентности интерфейса и UX в многоплатформенной разработке [2, 4].

**Functionize** использует NLP и ML для генерации «умных» тест-кейсов на основе требований и пользовательских сценариев, написанных на естественном языке. Платформа поддерживает автогенерацию и автоматическое обновление тестов, анализ причин сбоев, а также рекомендации по улучшению покрытия. Это облегчает внедрение ИИ в команды, не имеющие глубокого технического бэкграунда [2,5].

**Mabl** предлагает автоматизацию регрессионного тестирования с применением машинного обучения. Система анализирует результаты тестов, пользовательское поведение и метрики выполнения, выявляя нестабильные тесты и проблемные участки в приложении. Mabl также предлагает мощную интеграцию с CI/CD пайплайнами и DevOps-подходами [2,6].

Существуют и другие решения: **Sahi Pro**, **ACCELQ**, **Test.AI**, **Tricentis Tosca**, каждая из которых реализует различные аспекты интеллектуального тестирования. Общей чертой этих инструментов является ориентация на снижение зависимости от ручного ввода, повышение стабильности тестов и возможность обучения на производственных данных [7].

Важно отметить, что большинство современных платформ предоставляют возможности облачного хранения тестов, интеграции с системами управления проектами (Jira, GitLab, Jenkins) и отчётности в реальном времени, что делает их особенно привлекательными для крупных проектов с высокой степенью автоматизации [2].

**Преимущества применения ИИ в тестировании.** Интеграция ИИ в процессы тестирования открывает новые горизонты в обеспечении качества ПО. Ключевые преимущества технологии охватывают как технические аспекты, так и организационные выгоды [2,3].

**1. Сокращение времени на тестирование.** Один из наиболее ощутимых эффектов — резкое ускорение цикла тестирования. Генерация тест-кейсов, их запуск, анализ результатов и даже обновление сценариев могут выполняться без участия человека. Это особенно важно в условиях Agile и DevOps, где скорость релизов имеет решающее значение.

**2. Адаптивность.** В отличие от традиционных скриптов, ИИ-системы способны обучаться и адаптироваться к изменениям в приложении. Например, если интерфейс претерпел изменения, модель может определить, что произошло, и соответствующим образом обновить тест-кейсы. Это значительно сокращает затраты на сопровождение тестов [2].

**3. Повышение покрытия тестами.** ИИ-алгоритмы способны анализировать большое количество данных (истории коммитов, баг-репорты, пользовательское поведение), чтобы выявлять области с повышенным риском. Это позволяет строить более стратегические и эффективные тестовые наборы, минимизируя «мёртвые зоны» и увеличивая общее покрытие [2].

**4. Снижение человеческого фактора.** Поскольку ИИ берет на себя рутинные и повторяющиеся задачи, значительно уменьшается вероятность ошибок, связанных с невнимательностью, усталостью или субъективностью тестировщика. Это особенно актуально в стрессовых условиях или при выполнении больших объемов работы [2].

**5. Оптимизация регрессии.** ИИ может приоритизировать тесты, основываясь на вероятности возникновения дефектов. Это позволяет запускать в первую очередь наиболее критичные сценарии и экономить ресурсы [2].

**6. Интеллектуальный анализ логов и метрик.** Нейросети могут анализировать системные логи, определяя причины сбоев и их взаимосвязь с недавними изменениями. Это позволяет ускорить локализацию и исправление ошибок [4].

В результате, компании, внедрившие ИИ в QA, получают более надёжный, масштабируемый и предсказуемый процесс тестирования, что способствует улучшению качества конечного продукта и снижению затрат [2].

**Ограничения и вызовы.** Несмотря на значительные достижения и преимущества, автоматизация тестирования с использованием ИИ сталкивается с рядом сложных задач и ограничений, которые необходимо учитывать при её внедрении [2].

**1. Зависимость от качества данных.** Эффективность моделей ИИ напрямую зависит от объёма и качества данных, на которых они обучаются. Недостаток баг-репортов, нерепрезентативные данные или шум могут привести к некорректным выводам и снижению точности предсказаний [2].

**2. Проблема «чёрного ящика».** Большинство моделей глубокого обучения обладают высокой степенью сложности и непрозрачности. Это затрудняет интерпретацию их решений: почему тест провалился, как была выбрана стратегия тестирования — всё это может остаться непонятным для QA-инженера или заказчика [9].

**3. Высокие начальные затраты.** Внедрение ИИ требует вложений в инфраструктуру, обучение персонала, настройку пайплайнов CI/CD, а также в интеграцию с текущими системами. Для небольших команд это может быть критическим барьером [2,10].

**4. Ограниченная универсальность.** Хотя ИИ отлично работает в условиях повторяющихся и типичных сценариев, он может оказаться неэффективным при тестировании нестандартных решений, нестабильных интерфейсов или продуктов с высокой степенью новизны [2].

**5. Этические и юридические аспекты.** Использование ИИ может повлечь за собой вопросы ответственности за ошибки, особенно если сбой в системе произошёл из-за некорректных предсказаний ИИ. Кроме того, при использовании пользовательских данных возникает риск нарушения конфиденциальности [2].

**6. Поддержка нестабильных окружений.** Изменения в инфраструктуре (например, миграции баз данных или смена API) могут затруднить адаптацию ИИ-моделей, особенно если они были обучены на предыдущих конфигурациях [2].

Таким образом, для успешного внедрения ИИ в тестировании необходим взвешенный подход: тщательный анализ потребностей проекта, подготовка данных, выбор подходящих инструментов и постепенная интеграция с существующими процессами. В противном случае ожидаемый эффект может быть нивелирован техническими трудностями [2].

**Перспективы развития.** ИИ в автоматизации тестирования находится на этапе активного роста, и в ближайшие годы можно ожидать радикальных изменений в этой области. На передний план выходят следующие перспективные направления: [2].

**1. Применение генеративных моделей.** Современные модели вроде GPT и Codex уже способны писать тесты на основе описания требований. С дальнейшим развитием

генеративного ИИ можно ожидать автоматического создания целых тестовых наборов, адаптированных под различные платформы, языки и стандарты [2].

**2. Интеграция в CI/CD пайплайны.** ИИ всё активнее используется в DevOps-среде для создания самонастраивающихся пайплайнов, где тесты не просто запускаются автоматически, а выбираются и оптимизируются в зависимости от изменившегося кода. Это позволит добиваться высокой степени гибкости и минимального времени на проверку каждого коммита [2,10].

**3. Explainable AI (XAI).** Технологии объяснимого ИИ направлены на устранение проблемы «чёрного ящика». Развитие XAI поможет инженерам лучше понимать, почему ИИ принял то или иное решение, что повысит доверие к таким системам и облегчит их сертификацию в критически важных сферах [2,9].

**4. Автономные QA-агенты.** Создание ИИ-агентов, способных самостоятельно выявлять изменения в продукте, строить стратегию тестирования, проводить A/B тесты и даже предлагать улучшения — одно из самых амбициозных направлений. Такие агенты будут способны функционировать как полноценные участники команды разработки [2].

**5. Расширение мультимодального тестирования.** В будущем ИИ сможет проводить комплексное тестирование, объединяя текстовый, визуальный и голосовой интерфейсы, что особенно важно для мобильных и голосовых ассистентов [2].

**6. Этические и нормативные инициативы.** С ростом влияния ИИ в QA появится необходимость в формировании стандартов, регулирующих прозрачность, безопасность и воспроизводимость решений. Возможно, появление сертификаций ИИ-платформ по аналогии с ISO [2].

Эти и другие тренды свидетельствуют о том, что искусственный интеллект становится не просто вспомогательным инструментом, а ключевым элементом цифровой трансформации в сфере обеспечения качества [2].

Закключение. Автоматизация тестирования с использованием искусственного интеллекта (ИИ) представляет собой важный шаг в развитии современных методов обеспечения качества программного обеспечения. ИИ позволяет выйти за рамки традиционного тестирования, добавляя интеллектуальные возможности к процессам, которые ранее были ограничены ручными сценариями или рутинной автоматизацией [2].

Внедрение таких технологий, как машинное обучение, нейронные сети и обработка естественного языка, позволяет создавать адаптивные и самонастраивающиеся тестовые системы, способные анализировать поведение пользователей, изменяющийся код, а также автоматически генерировать и оптимизировать тестовые сценарии. Это открывает путь к повышению точности, скорости и надёжности процессов QA в условиях ускоряющихся релизных циклов и постоянных изменений [2].

Однако полное внедрение ИИ в тестирование сопряжено с рядом вызовов, включая необходимость в качественных данных, проблемы интерпретируемости моделей, высокую стоимость внедрения, а также потенциальные риски, связанные с этическими и юридическими аспектами использования ИИ. Эти ограничения требуют стратегического подхода: важно не просто использовать технологии ради тренда, а интегрировать их там, где они действительно приносят ценность, усиливая возможности человека, а не подменяя его [2].

В то же время перспективы развития ИИ в тестировании выглядят многообещающе. Генеративные модели, автономные агенты, XAI и интеграция в DevOps-процессы — всё это уже перестаёт быть теорией и становится частью промышленной практики. Мы стоим на пороге новой эры обеспечения качества, где ИИ играет не вспомогательную, а ведущую роль [2,10].

Таким образом, можно заключить, что ИИ станет неотъемлемым элементом инструментального арсенала будущего QA-инженера. Его возможности позволят компаниям создавать более стабильные, надёжные и ориентированные на пользователя программные продукты, одновременно снижая затраты и повышая производительность процессов разработки и тестирования [2].

## СПИСОК ЛИТЕРАТУРЫ

1. Башили, В. Р., Ромбах, Х. Д. Принципы программной инженерии [Текст] / пер. с англ. — М.: Вильямс, 2020. — 384 с.
2. Хейст, Д., Рич, Ч., Найт, К. Искусственный интеллект. Современный подход [Текст] / пер. с англ. — М.: Вильямс, 2007. — 864 с.
3. Канер, К., Бах, Д., Петтикорд, Б. Уроки тестирования программного обеспечения [Текст] / пер. с англ. — М.: Вили-Пресс, 2003. — 320 с.
4. Лавров, А. И., Боровков, А. И. Машинное обучение и нейронные сети. [Текст] — СПб.: Питер, 2021. — 448 с.
5. Поляков, А. В. Автоматизированное тестирование программных систем. [Текст] М.: Горячая линия – Телеком, 2018. — 272 с.
6. Кобринский, А. И. Системы искусственного интеллекта. [Текст] М.: Радио и связь, 2003. — 320 с.
7. Глушков, В. М. Введение в автоматизацию и кибернетику. [Текст] М.: Наука, 1987. — 464 с.
8. Золотарев, А. М. Тестирование программ: теория и практика. [Текст] СПб.: БХВ-Петербург, 2019. — 400 с.
9. Столяр, С. А. Обеспечение качества программного обеспечения. [Текст] М.: Форум, 2015. — 288 с.
10. Самойлов, А. И. Интеллектуальные системы управления и принятия решений. [Текст] — М.: Машиностроение, 2011. 352 с.

## REFERENCES

1. Basili, V. R., Rombah, H. D. Principy programmnoj inzhenerii [Principles of software engineering]. (2020): - (In Rus)
2. Hejst, D., Rich, Ch., Najt, K. Iskusstvennyj intellekt. Sovremennyj podhod [Artificial intelligence. A modern approach] // (2007): - (In Rus)
3. Kaner, K., Bah, D., Pettikord, B. Uroki testirovaniya programmnoho obespecheniya [Software Testing Lessons] // (2003): - (In Rus)
4. Lavrov, A. I., Borovkov, A. I. Mashinnoe obuchenie i nejronnye seti. [Machine learning and neural networks.]. (2021): - (In Rus)
5. Polyakov, A. V. Avtomatizirovannoe testirovanie programmnyh sistem. [Automated testing of software systems.]. (2018): - (In Rus)
6. Kobrinskij, A. I. Sistemy iskusstvennogo intellekta. [Artificial intelligence systems.]. (2003): - (In Rus)
7. Glushkov, V. M. Vvedenie v avtomatizaciyu i kibernetiku. [Introduction to automation and cybernetics.]. (1987): - (In Rus)
8. Zolotarev, A. M. Testirovanie programm: teoriya i praktika. [Software testing: theory and practice.]. (2019): - (In Rus)
9. Stolyar, S. A. Obespechenie kachestva programmnoho obespecheniya. [Software quality assurance.]. (2015): - (In Rus)

10. Samojlov, A. I. Intellektual'nye sistemy upravleniya i prinyatiya reshenij. [Intelligent management and decision-making systems.]. (2011): - (In Rus)

### ЖАСАНДЫ ИНТЕЛЛЕКТТИ ҚОЛДАНУҒА НЕГІЗДЕЛГЕН АВТОМАТТАНДЫРЫЛҒАН ТЕСТІЛЕУ

**Аңдатпа.** Бұл мақалада бағдарламалық жасақтаманы тестілеу үдерісін автоматтандыруда жасанды интеллект (ЖИ) технологияларын қолдану мәселесі қарастырылады. Бағдарламалық жүйелердің күрделенуі мен өнімді жеткізу мерзімдерінің қысқаруына байланысты дәстүрлі тестілеу әдістерінің тиімділігі төмендеп келеді. Жасанды интеллектті қолдану тестілеу үдерістерін оңтайландырып, олардың икемділігі мен үнемділігін арттыруға жаңа мүмкіндіктер ашады. Мақалада машиналық оқыту, нейрондық желілер, табиғи тілді өңдеу және шешім қабылдаудың интеллектуалды жүйелері секілді негізгі технологиялар талданады. Сондай-ақ, Testim.io, Applitools, Functionize және Mabl сияқты ЖИ қолданатын заманауи тестілеу құралдарына шолу жасалады. ЖИ-ді QA жүйесіне енгізудің басты артықшылықтары ретінде тесттерді автоматты түрде генерациялау, өнім өзгерістеріне бейімделу, күрделі қателерді анықтау және қамтуды кеңейту атап өтіледі. Сонымен қатар, «қара жәшік» мәселесі, сапалы деректерге деген сұраныс, енгізу құны мен тұрақсыз интерфейстерді қолдау қиындықтары секілді шектеулер де қарастырылады. Қорытындыда генеративті модельдерді қолдану, Explainable AI және автономды тестілеуші агенттерді дамыту секілді болашағы зор бағыттар сипатталады.

**Кілт сөздер:** жасанды интеллект, тестілеуді автоматтандыру, машиналық оқыту, нейрондық желілер, интеллектуалды жүйелер, бағдарламалық жасақтаманы тестілеу, ЖИ QA-де.

### AUTOMATED TESTING BASED ON ARTIFICIAL INTELLIGENCE

**Abstract.** This paper explores the application of artificial intelligence (AI) technologies in the automation of software testing processes. With the increasing complexity of software systems and the shortening of product delivery timelines, traditional testing approaches are becoming less effective. The use of AI opens up new opportunities to optimize testing processes, improve flexibility, and reduce costs. The article analyzes key technologies such as machine learning, neural networks, natural language processing, and intelligent decision-making systems. It also provides an overview of modern AI-based testing tools, including Testim.io, Applitools, Functionize, and Mabl. The main advantages of integrating AI into QA include automatic test generation, adaptability to changes, detection of complex defects, and expanded coverage. At the same time, certain challenges are discussed, such as the "black box" problem, the demand for high-quality data, implementation costs, and the difficulty of supporting unstable interfaces. The conclusion outlines promising directions such as the use of generative models, Explainable AI, and the development of autonomous testing agents. The analysis shows that the integration of AI into testing can make the process more intelligent, efficient, and adaptive, with great potential for the future.

**Keywords:** artificial intelligence, test automation, machine learning, neural networks, intelligent systems, software testing, AI in QA.